WHAT IS CLAIMED IS:

1. A method for scheduling arriving data packets for input to a switch having a plurality of input channels, and a plurality of output channels, said scheduling being performed in successive scheduling phases where each scheduling phase further comprises at least log N scheduling iterations, the method comprising the steps of:

prior to a first scheduling iteration of each scheduling phase:

receiving at each of said plurality of input channels data packets destined for transmission to one of said plurality of output channels and wherein each of said plurality of input and output channels are classified as unmatched prior to said first scheduling iteration; and

storing said received data packets into at least one data scheduling envelope associated with each of said plurality of input channels, said at least one data scheduling envelope being configured to store a plurality of data packets;

in each of said at least log N scheduling iterations of said each scheduling phase:

(a) assigning a channel pair weight to unmatched input-unmatched output channel pairs having a data scheduling envelope storing at least one data packet destined for transmission;

(b) sending a match request from each unmatched input channel to an unmatched output channel having a highest channel pair weight there-between; and

(c) accepting a match request at each unmatched output channel from an unmatched input channel whose assigned channel pair weight there-between is determined to be

highest from among all received match requests.

2. The method of claim 1 wherein the step of assigning a channel pair weight to unmatched input-unmatched output channel pairs further includes the steps of:

for each unmatched input-output channel pair:

determining a number of bytes to be transmitted there-between;

determining a delay value associated with a received data packet having a highest time-stamp to be transmitted there-between; and

computing an assigned channel pair weight as a linear combination of said determined number of bytes and said determined delay.

3. A method for scheduling arriving data packets for input to a switch having a plurality of input channels, i=1...N, and a plurality of output channels, j=1..N, said scheduling being performed in successive scheduling phases where each scheduling phase further comprises at least log N scheduling iterations, each of said plurality of input channels being further comprised of N virtual output queues, VOQs (i,j), for buffering at least one data packet P(i,j) received on the Ith input channel destined for the Jth output channel, the method comprising the steps of:

in each scheduling phase:

receiving said at least one data packet, P(i,j), received at said Ith input channel destined for said jth output channel;

storing said at least one received data packet, P(i,j), in an ith channel data scheduling envelope associated with a virtual output queue VOQ(i,j), wherein a first scheduling envelope at said VOQ(i,j) is a head-of-line data scheduling envelope and a first stored data packet in said head-of-line data scheduling envelope is a head-of-line packet;

in each of said at least log N scheduling iterations of each scheduling phase:

(a) assigning a weight, w_{ij}, to each non-empty head-of-line scheduling envelope for each input channel unmatched in a previous log N scheduling iteration;

(b) sending a match request from each unmatched input channel to an unmatched output channel whose assigned weight w_{ij} there-between is highest;

(c) determining at each of said unmatched output channels a highest assigned weight w_{ij} from among all received input channel match requests; and

(d) granting at each of said unmatched output channels a match request to an input channel whose determined assigned weight w_{ij} there-between is highest.

4. The method of claim 3 wherein the step of assigning a weight to each non-empty virtual output queue VOQ(i,j) for each ith input channel not matched in a previous scheduling iteration further comprises the step of computing said assigned weight as:

$$w_{ij} = [\log_2 (q_{ij}) + 1] - K$$

where q_{ij} is the number of bytes arriving at an input channel from among said plurality of input channels destined for output channel from among said plurality of output channels , and K is the logarithm of the number of bytes in a smallest received packet.

5. The method of claim 3 wherein the step of assigning a weight to each non-empty virtual output queue VOQ(i,j) for each Ith input channel not matched in a previous scheduling iteration further comprises the steps of:

determining a delay d_{ij} as a number of said at least log N scheduling iterations which have passed, as measured from a point at which a head-of-line (HOL) data packet is received in a HOL envelope at said non-empty virtual output queue VOQ(i,j);

assigning a maximum weight $w_{ij}(max)$ to said virtual output queue VOQ(i,j) if said delay d_{ij} exceeds a predetermined threshold; and

otherwise, computing said assigned weight as:

$$w_{ij} = [\log_2 (q_{ij}) + 1] - K$$

where q_{ij} is the number of bytes arriving at input channel destined for output channel, and K is the logarithm of the number of bytes in a smallest received packet.

6. The method of claim 3 wherein the step of assigning a weight to each non-empty virtual output queue VOQ(i,j) for each Ith input channel not matched in a previous scheduling iteration further comprises the step of computing said assigned weight as:

$$w_{ij} = [\log_2(d_{ij})] + K'$$

where d_{ij} is the delay of a head-of-line (HOL) data packet in a HOL envelope at VOQ(i,j), and $K' = [\log_2[E] - K$

where

E is the number of of bytes per envelope, and

K is the logarithm of the number of bytes in a smallest received packet.

7. The method of claim 3 wherein the step of assigning a weight to each non-empty virtual output queue VOQ(i,j) for each Ith input channel not matched in a previous scheduling iteration further comprises the step of computing said assigned weight as:

$$w_{ij} = [\log_2(d_{ij})] + [\log_2(e_{ij}) - K]$$

where

e_{ij} is the number of bytes in a head-of-line (HOL) envelope;

d_{ij} is the delay of a HOL data packet in a HOL envelope at VOQ(i,j); and

K is the logarithm of the number of bytes in a smallest received packet.

8. Apparatus for scheduling arriving data packets for input to a switch having a plurality of input channels and output channels, each of said plurality of input channels being further comprised of J virtual output queues, VOQs (i,j), for buffering said data packets received on the Ith input channel destined for the jth output channel, DP(i,j), the apparatus comprising:

means, associated with each input channel, for storing the data packets, DP(i,j), arriving at the associated input channel in at least one scheduling envelope residing at an associated VOQ(i,j);

means, associated with each input channel, for assigning a weight, W_{ij}, to each

VOQ(i,j) having at least one non-empty scheduling envelope;

means, associated with each input channel, having at least one VOQ(i,j) having an assigned weight, W_{ij}, for sending a request to the jth output having the highest assigned weight W_{ij}; and

means, associated with each output channel J, for granting a request from an output channel J, responsive to said received input channel requests, to an input channel I having the highest assigned weight, W_{ij}.

9. The apparatus of claim 8, further including:

means, associated with each output channel, for matching an input-output channel pair wherein the input-output channel pair has the highest assigned weight, W_{ij}, there-between.

10. The apparatus of claim 8, wherein the means for assigning a weight to each VOQ(i,j) further includes,

means, associated with each VOQ(i,j), for determining a total number of bytes comprising said stored data packets arriving at the associated VOQ(i,j); and

means, associated with each VOQ(i,j), for determining a delay associated with a head-of-line packet.

11. A switching apparatus for distributing data packets input from each of a plurality of input ports to a plurality of output ports, said switching apparatus comprising:

a comparator connected to simultaneously receive weight elements in successive clock cycle iterations, said comparator configured to output a largest weight element from among the received weight elements in each clock cycle iteration;

a demultiplexer configured to receive said largest weight element from said comparator, said demultiplexer further configured to distribute said largest weight elements to output lines; and

a plurality of comparators each configured to receive the largest weight elements from a corresponding output line of said output lines, each of said plurality of output comparators being further configured to output a largest weight element from among said received weight elements.

12. A switching apparatus for distributing data packets input from each of a plurality of input ports to a plurality of output ports, said switching apparatus comprising:

means for simultaneously receiving weight elements in successive clock cycle iterations;

means for outputting a largest weight element from among the received weight elements in each clock cycle iteration;

means for distributing said largest weight element from said outputting means; and

output selection means for selecting the overall largest weight element from among a plurality of largest weight elements received from said distributing means.